Activation based XAI Methods

2023.02.02

Jungmin Kim
KAIST Graduate School of AI

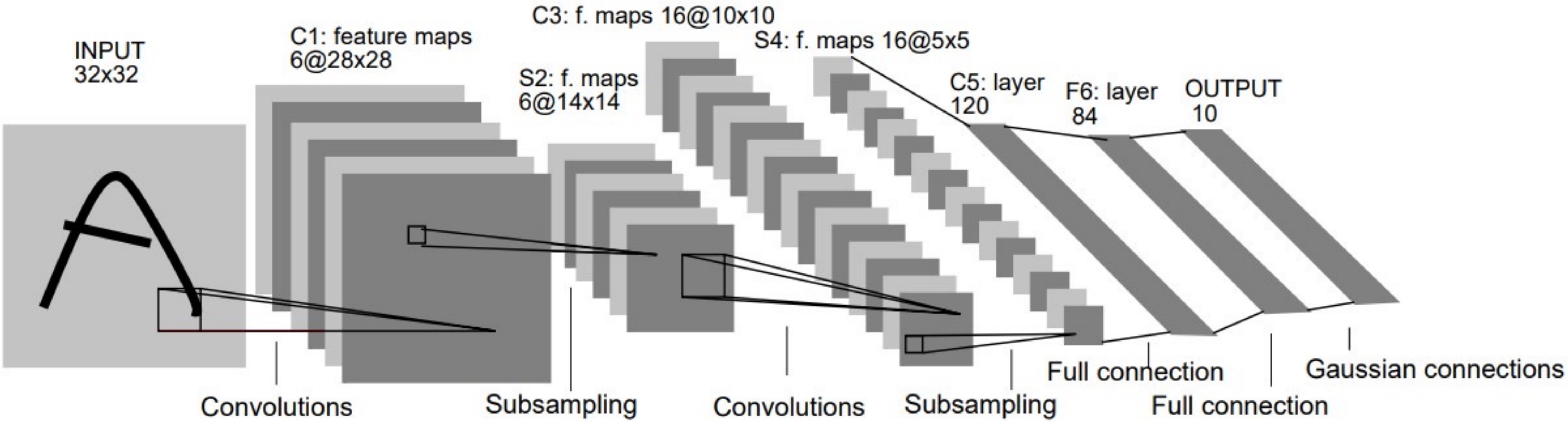
eXplainable Artificial Intelligence Center

KAIST XAI Tutorial Series
2023. 1. 26 – 2. 16

Contents

1. CNN and Activation Maps

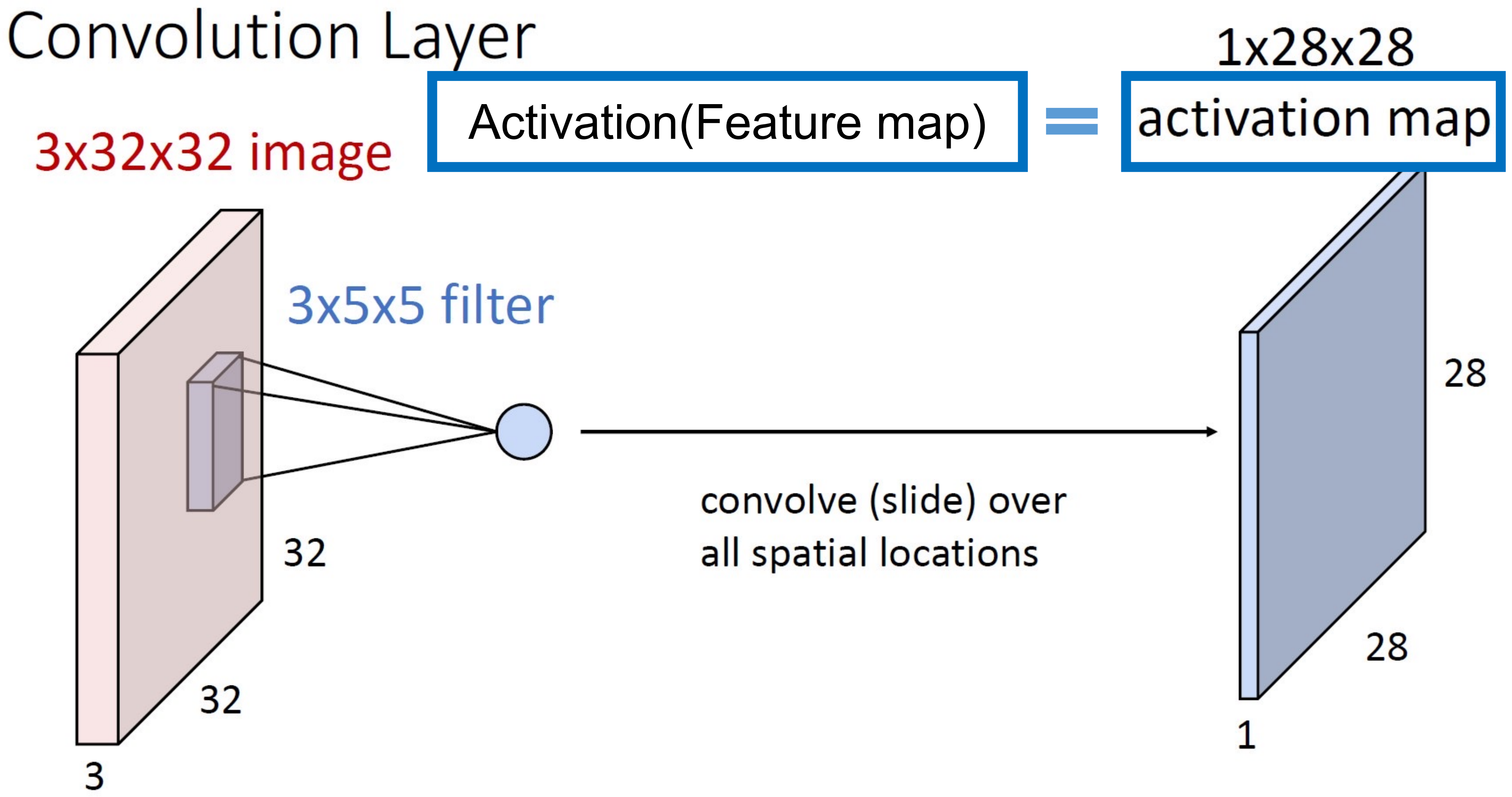
Convolutional Neural Networks (CNN)



- Components of a Convolutional Neural Networks
  - Convolutional layers
  - Activation function
  - Fully-Connected layers
  - Pooling layers

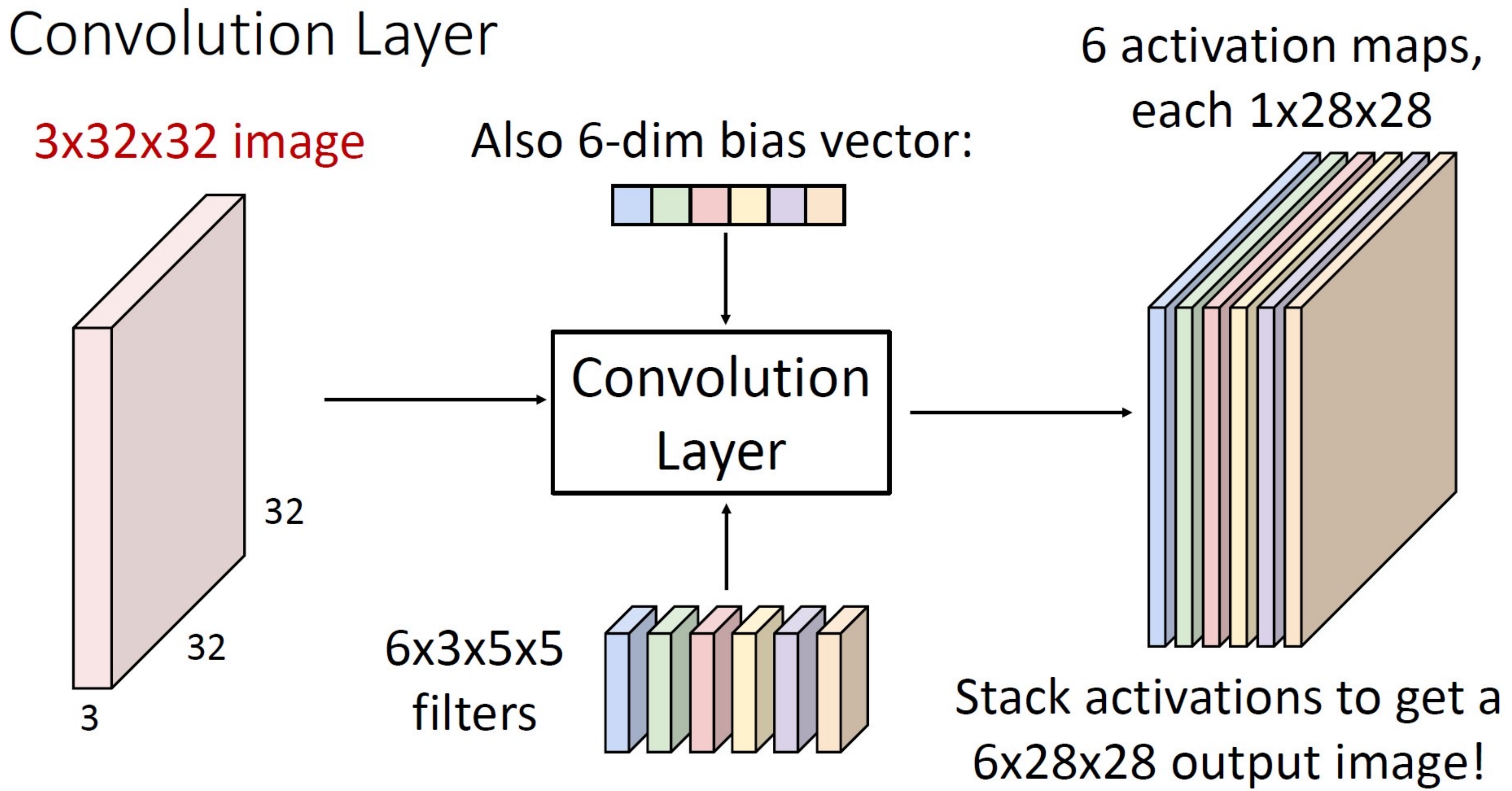
LeCun et al. (1998), GradientBased Learning Applied to Document

Activation Maps



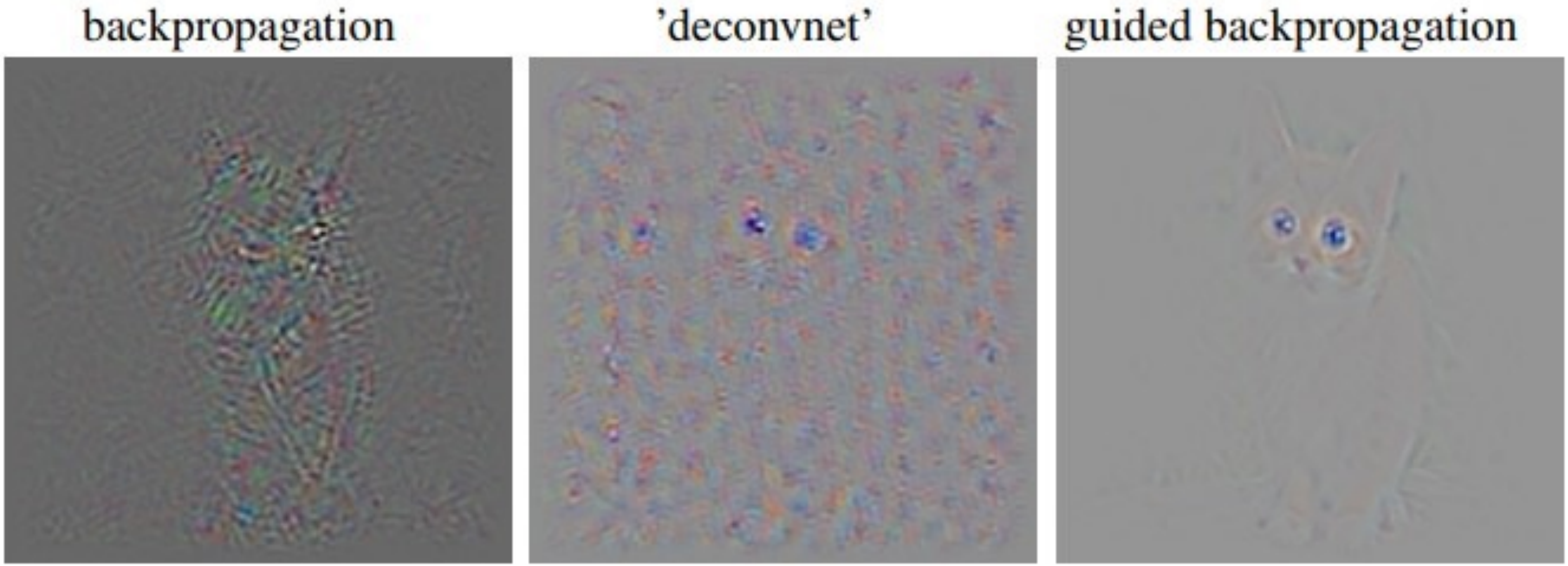
University of Michigan, EECS 498/598: Deep Learning for Computer Vision

Activation Maps



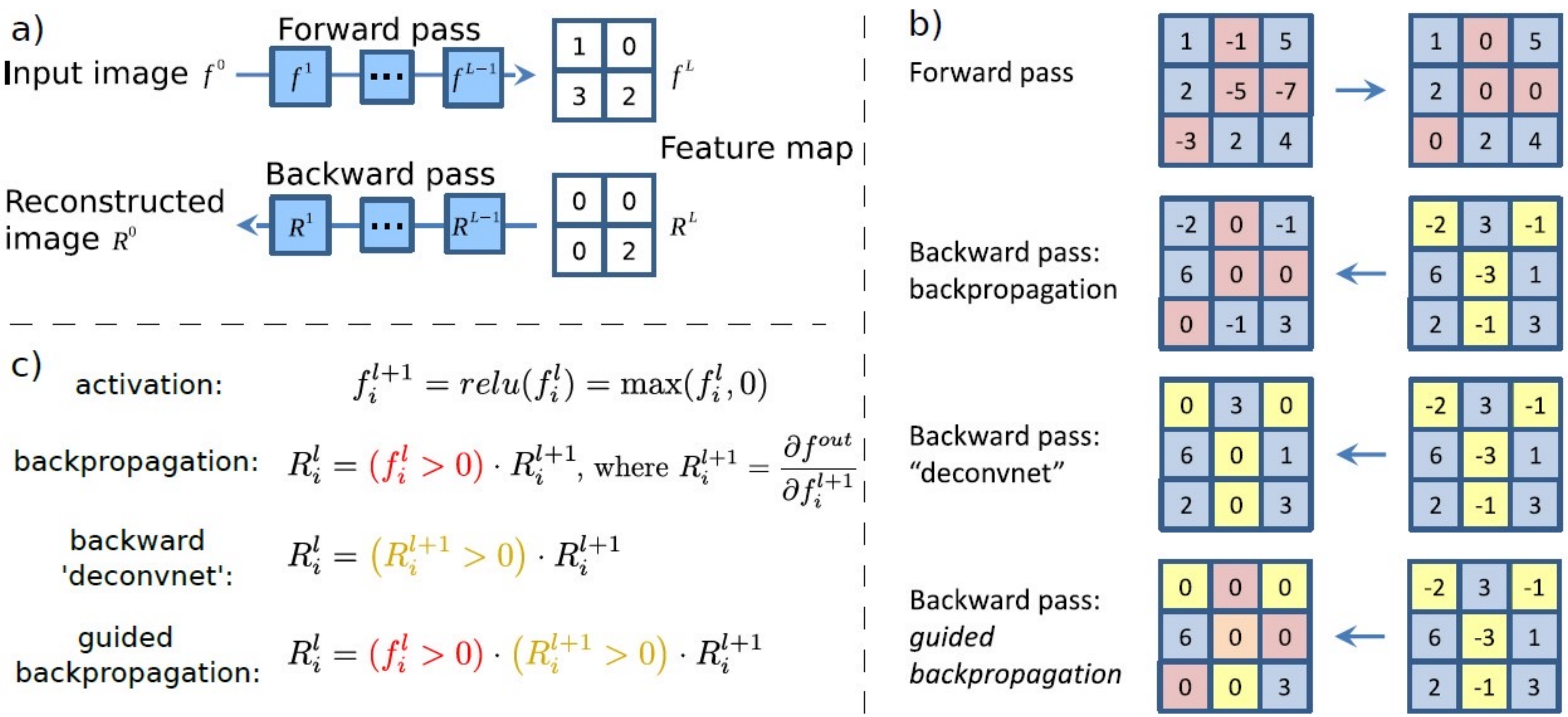
University of Michigan, EECS 498/598: Deep Learning for Computer Vision

Previous gradient based methods



Springenber et al. (2014), STRIVING FOR SIMPLICITY:THE ALL CONVOLUTIONAL NET

Previous gradient based methods



- Backpropagation
- Deconvnet
- Guided Backpropagation

Springenber et al. (2014), STRIVING FOR SIMPLICITY:THE ALL CONVOLUTIONAL NET

Limitation of previous methods

- No Class-discriminative
- Not fully utilizing the CNN's localization ability
- Only analyzing the convolutional layers, ignoring FC layers

Springenber et al. (2014), STRIVING FOR SIMPLICITY:THE ALL CONVOLUTIONAL NET

2. Class Activation Maps

Class Activation Maps (CAM)

Paper: Learning Deep Features for Discriminative Localization

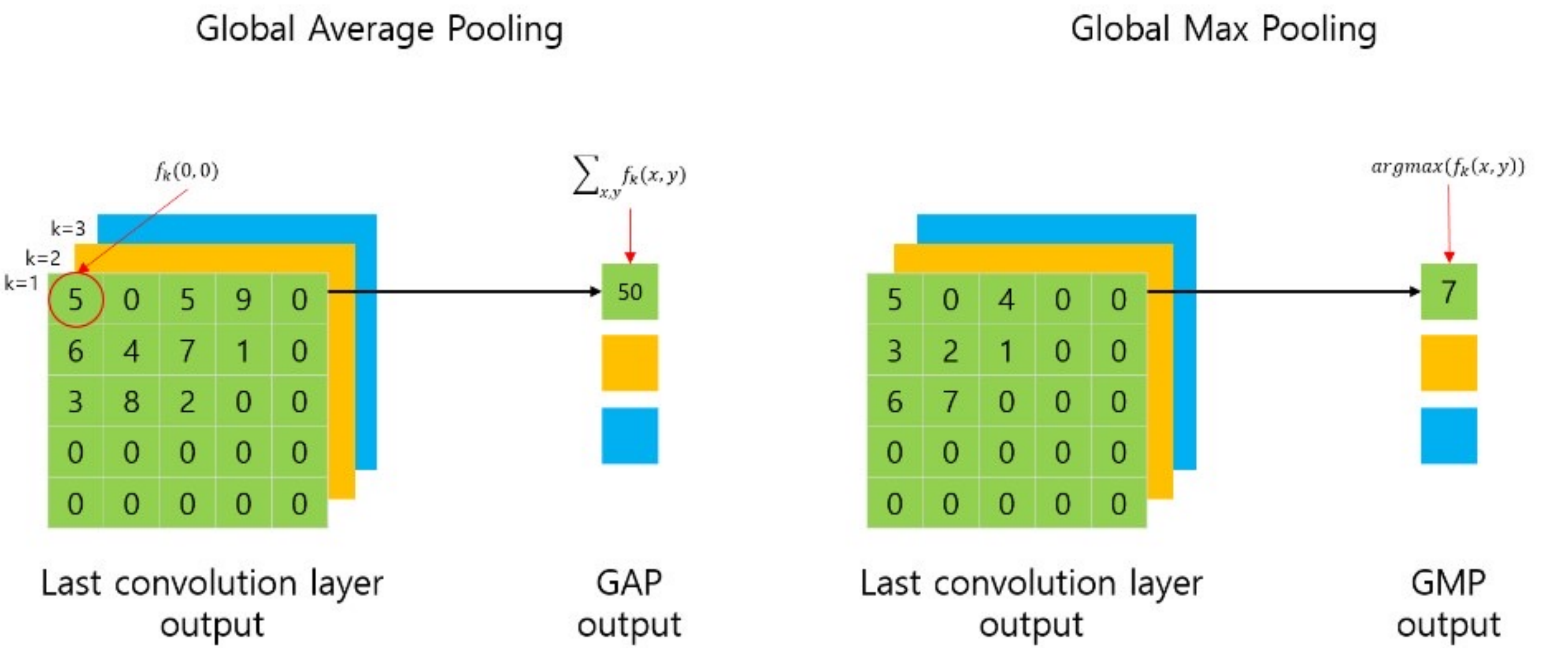
CAM

- CAM allows classification-trained CNN to both classify the image and
- Localize class specific image regions in a single forward pass

B Zhou et al. (2015), Learning Deep Features for Discriminative Localization

Class Activation Maps (CAM)

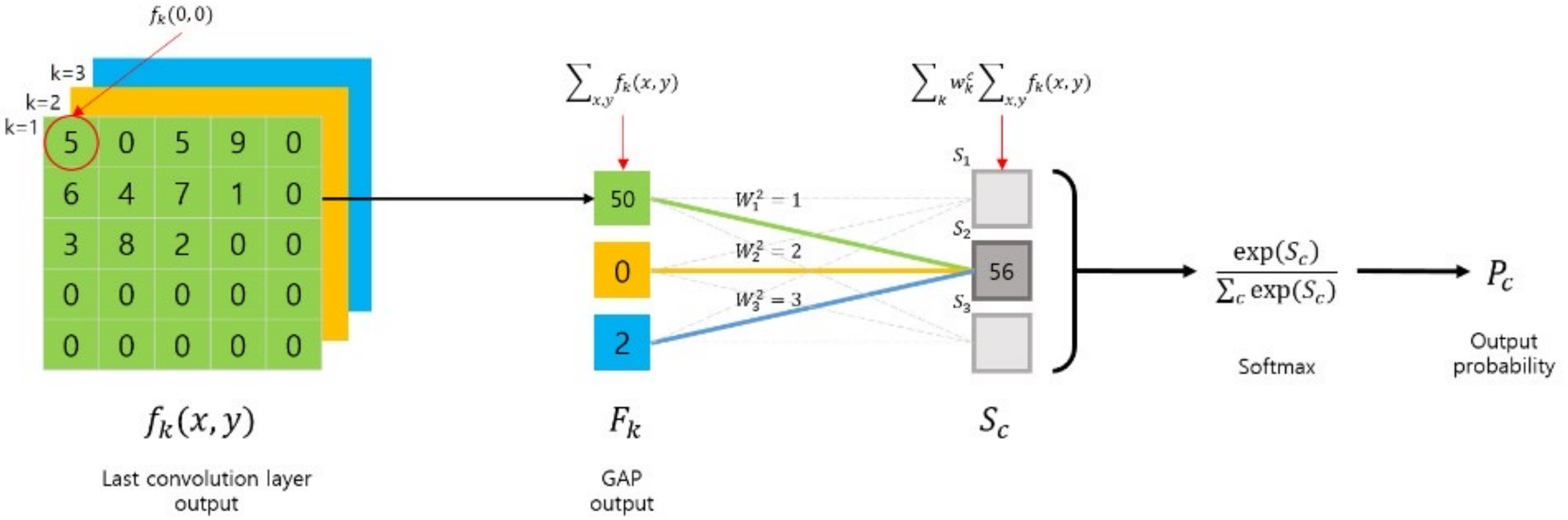
- How to express localizable deep representation?
  - Utilizing GAP layer



https://you359.github.io/cnn visualization/CAM/

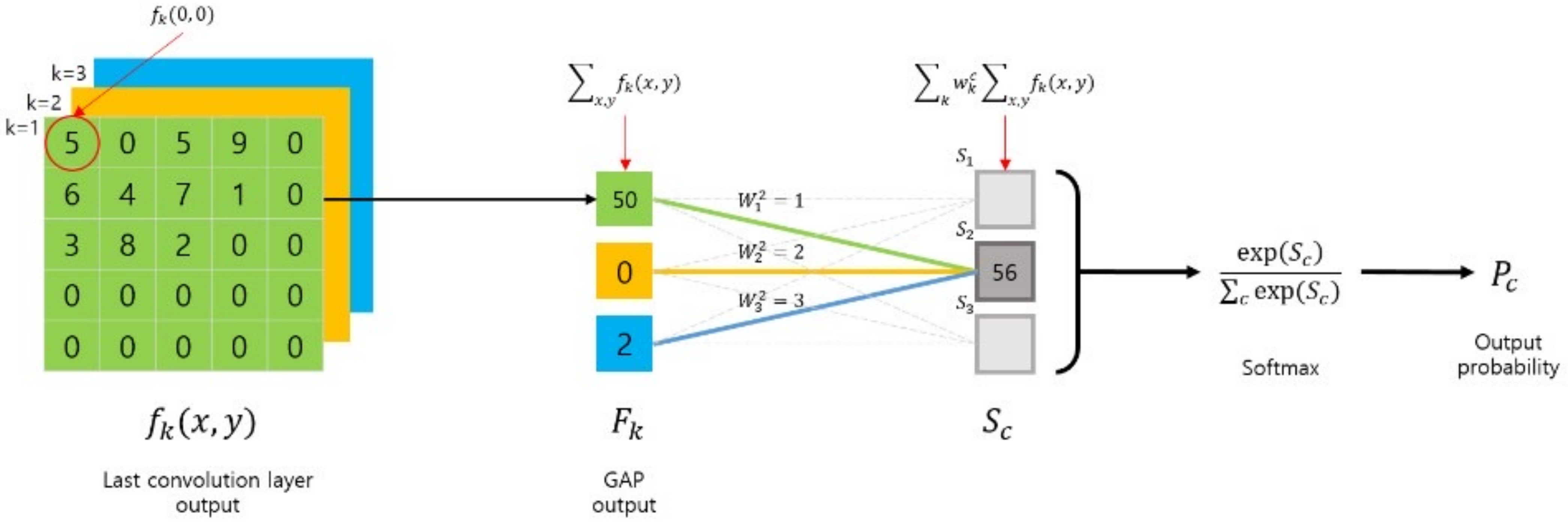
Class Activation Maps (CAM)

- Replacing (FC layers) to (GAP layers + FC layers)
  - Decreasing the number of parameters
  - GAP layers keep the spatial information, then better for localization



https://you359.github.io/cnn visualization/CAM/

Class Activation Maps (CAM)



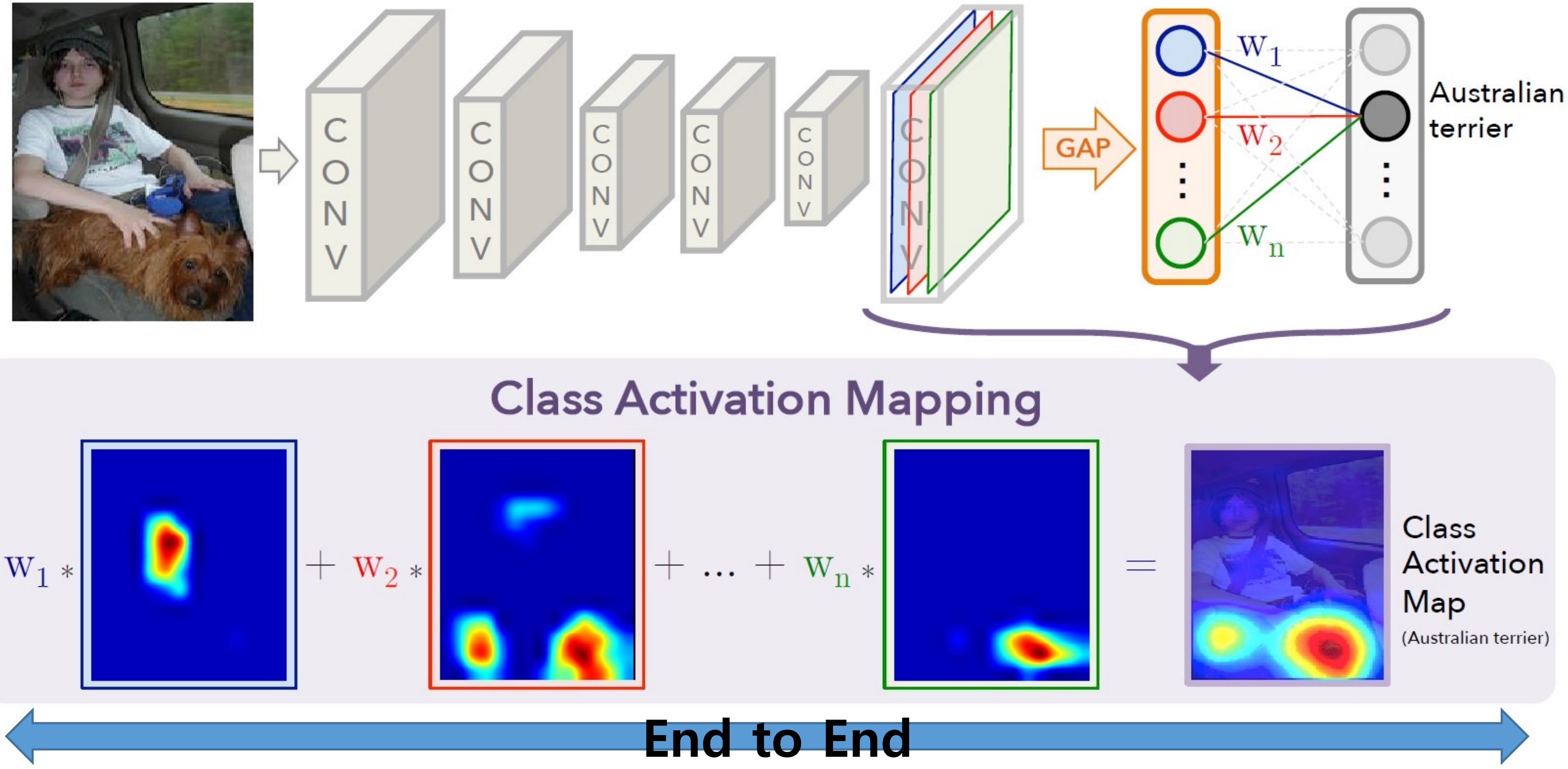
F^k : GAP output of Kth feature map, $\sum_{x,y} f_k(x,y)$

w_k^c : the importance of F^k for class c

$$S_c = \sum_k w_k^c \sum_{x,y} f_k(x,y) = \sum_{x,y} \sum_k w_k^c f_k(x,y).$$

https://you359.github.io/cnn visualization/CAM/

Class Activation Mapping



B Zhou et al. (2015), Learning Deep Features for Discriminative Localization

CAM Results

- Class-discriminative Localization

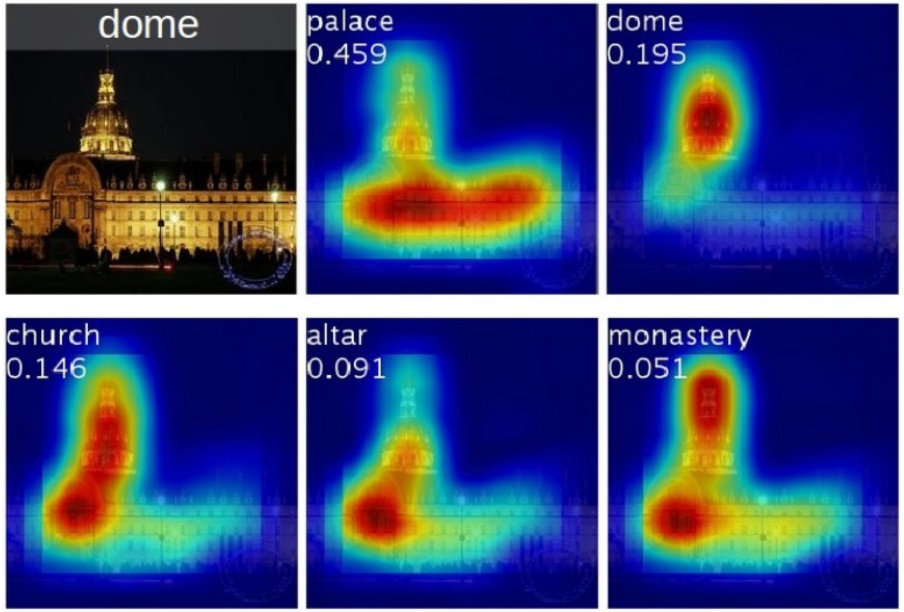


Figure 4. Examples of the CAMs generated from the top 5 predicted categories for the given image with ground-truth as dome. The predicted class and its score are shown above each class activation map. We observe that the highlighted regions vary across predicted classes e.g., *dome* activates the upper round part while *palace* activates the lower flat part of the compound.

- Compare with backprop methods – maps /localization

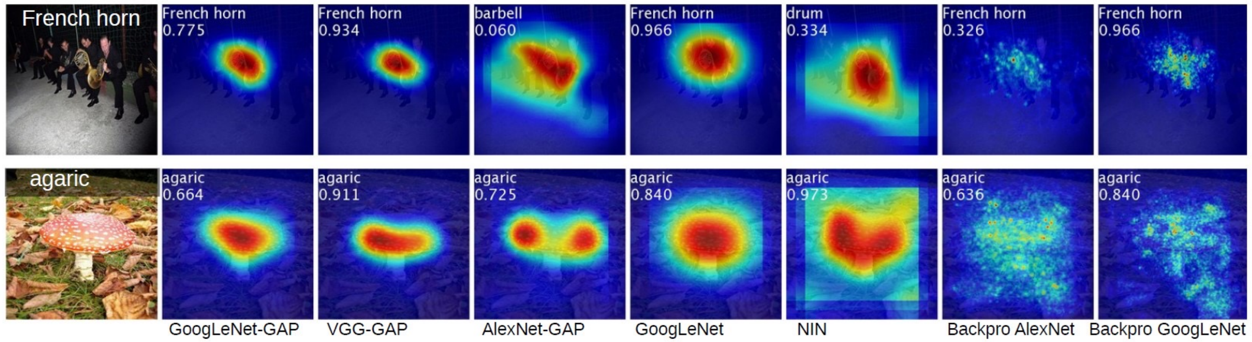


Figure 5. Class activation maps from CNN-GAPs and the class-specific saliency map from the backpropagation methods.

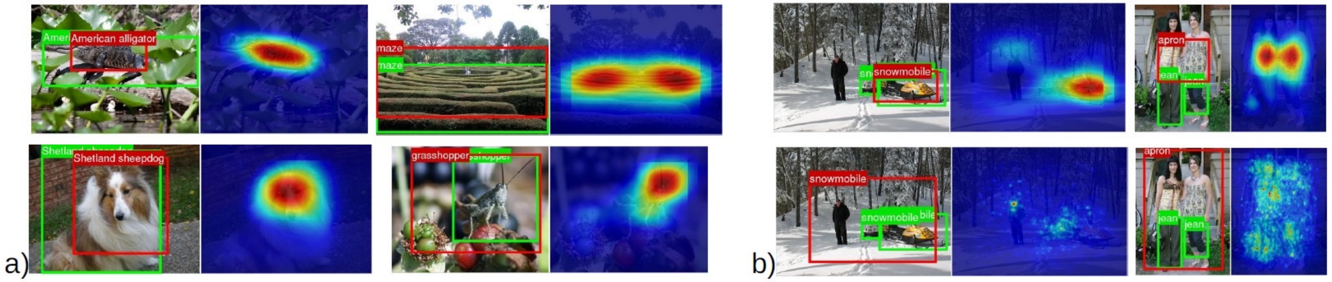


Figure 6. a) Examples of localization from GoogleNet-GAP. b) Comparison of the localization from GooleNet-GAP (upper two) and the backpropagation using AlexNet (lower two). The ground-truth boxes are in green and the predicted bounding boxes from the class activation map are in red.

B Zhou et al. (2015), Learning Deep Features for Discriminative Localization

3. Grad-CAM

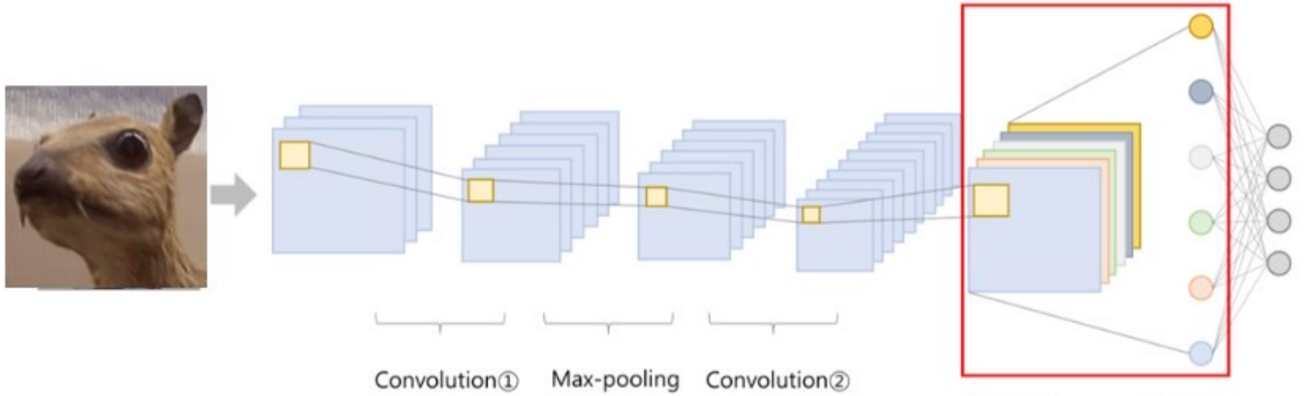
Limitation of CAM / Grad-CAM

- Architectural limitation of CAM
  - Conv feature maps > GAP > Softmax layer
  - Performance loss caused by using GAP
  - Inapplicable to any other tasks (Image captioning or VQA)
- Grad-CAM solve the limitation
  - Not require any modification in the network architecture (Not need GAP)
  - Applicable CNNs with FCs / CNN used for structured outputs / CNN used in task with multi-modal inputs

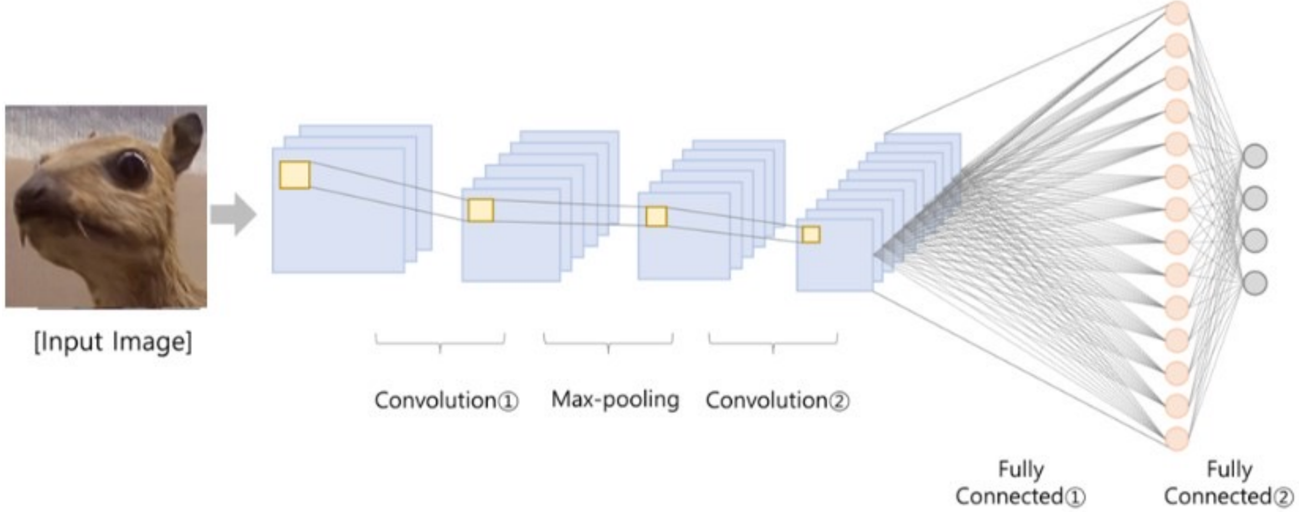
RR Selvaraju et al. (2016), Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

CAM <> Grad-CAM

CAM



Grad-CAM



https://you359.github.io/cnn visualization/CAM/

Grad-CAM

$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i \sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}}$$

- First, calculate the gradient of the score for class c with respect to feature map activations
- Then, global-average-pooled
- The weight for feature map is calculated by gradients, not by learning

$$L_{\text{Grad-CAM}}^c = ReLU \underbrace{\left(\sum_k \alpha_k^c A^k\right)}_{\text{linear combination}}$$

- α_k^c : the importance of feature map k for a target class c
- ReLU: only a positive influence on the class of interest

RR Selvaraju et al. (2016), Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

Grad-CAM generalizes CAM

CAM

$$Y^c = \sum_k \underbrace{w_k^c}_{\text{class feature weights}} \overbrace{\frac{1}{Z}\sum_i \sum_j}^{\text{global average pooling}} \underbrace{A_{ij}^k}_{\text{feature map}} \qquad (3)$$

Let us define F^k to be the global average pooled output,

$$F^k = \frac{1}{Z}\sum_i \sum_j A_{ij}^k \qquad (4)$$

CAM computes the final scores by,

$$Y^c = \sum_k w_k^c \cdot F^k \qquad (5)$$

where w_k^c is the weight connecting the k^{th} feature map with the c^{th} class. Taking the gradient of the score for class c (Y^c) with respect to the feature map F^k we get,

$$\frac{\partial Y^c}{\partial F^k} = \frac{\frac{\partial Y^c}{\partial A_{ij}^k}}{\frac{\partial F^k}{\partial A_{ij}^k}} \qquad (6)$$

Taking partial derivative of (4) w.r.t. A_{ij}^k, we can see that $\frac{\partial F^k}{\partial A_{ij}^k} = \frac{1}{Z}$. Substituting this in (6), we get,

$$\frac{\partial Y^c}{\partial F^k} = \frac{\partial Y^c}{\partial A_{ij}^k} \cdot Z \qquad (7)$$

From (5) we get that, $\frac{\partial Y^c}{\partial F^k} = w_k^c$. Hence,

$$w_k^c = Z \cdot \frac{\partial Y^c}{\partial A_{ij}^k} \qquad (8)$$

Summing both sides of (8) over all pixels (i, j),

$$\sum_i \sum_j w_k^c = \sum_i \sum_j Z \cdot \frac{\partial Y^c}{\partial A_{ij}^k} \qquad (9)$$

Since Z and w_k^c do not depend on (i, j), rewriting this as

$$Z w_k^c = Z \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k} \qquad (10)$$

Note that Z is the number of pixels in the feature map (or $Z = \sum_i \sum_j \mathbf{1}$). Thus, we can re-order terms and see that

CAM

$$w_k^c = \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k} \qquad (11)$$

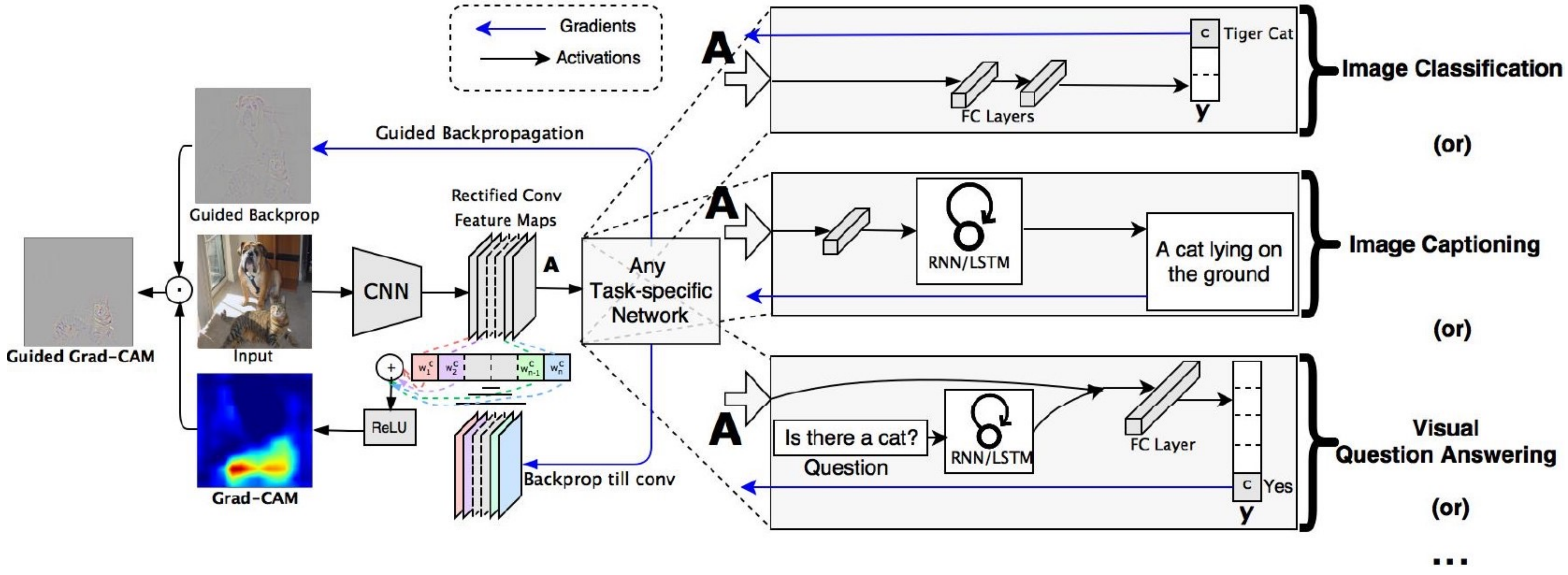
- 1/Z : a proportionality constant
- w_k^c is identical to α_k^c used by Grad-CAM

Grad-CAM

$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i \sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}}$$

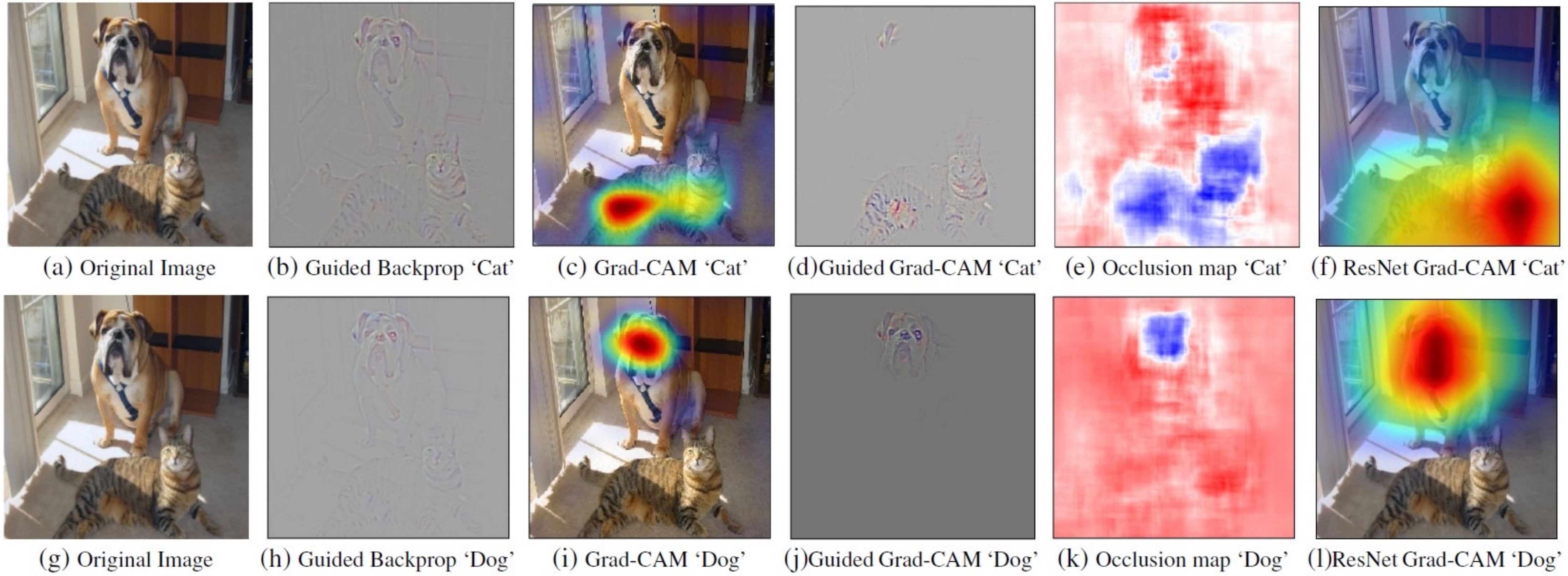
RR Selvaraju et al. (2016), Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

Grad-CAM / Guided Grad-CAM



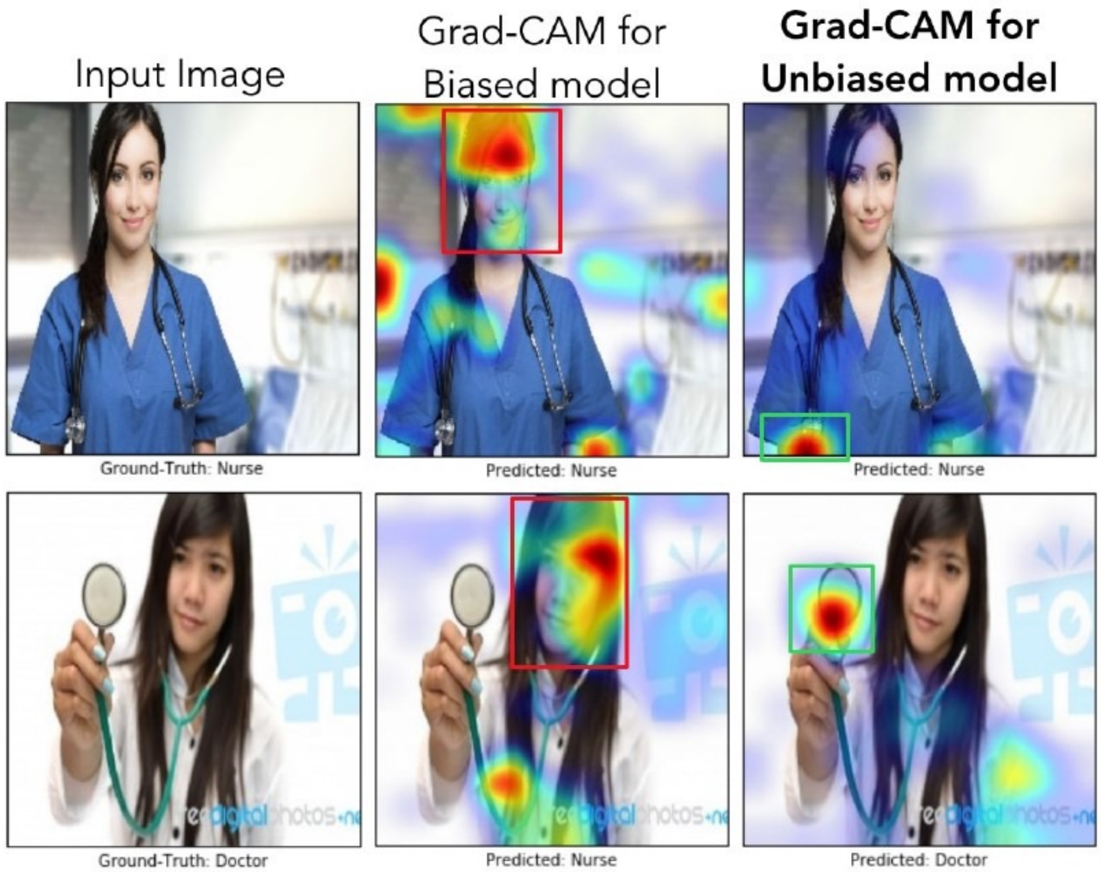
RR Selvaraju et al. (2016), Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

Grad-CAM Results

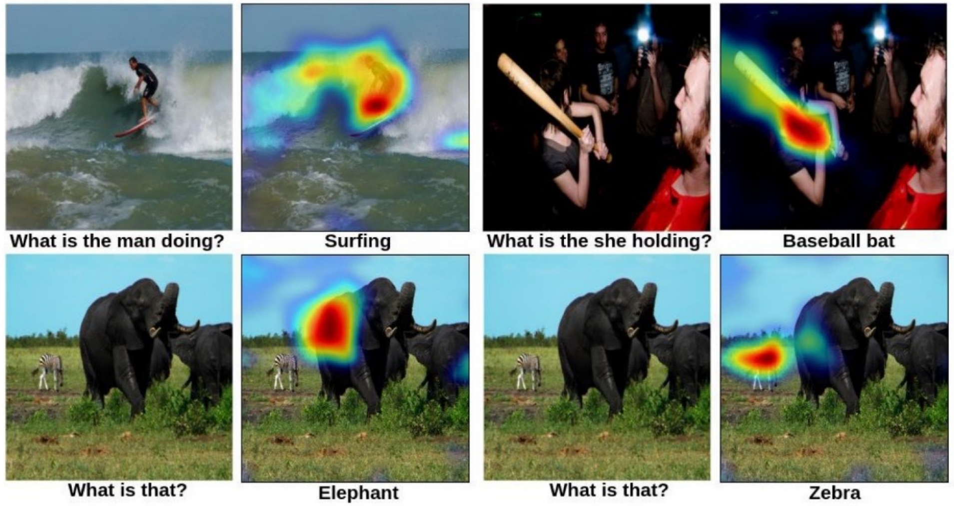


RR Selvaraju et al. (2016), Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

Grad-CAM Results



(a) Image captioning explanations



(b) Visualizing ResNet based Hierarchical co-attention VQA model from [39]

RR Selvaraju et al. (2016), Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

4. Grad-CAM++

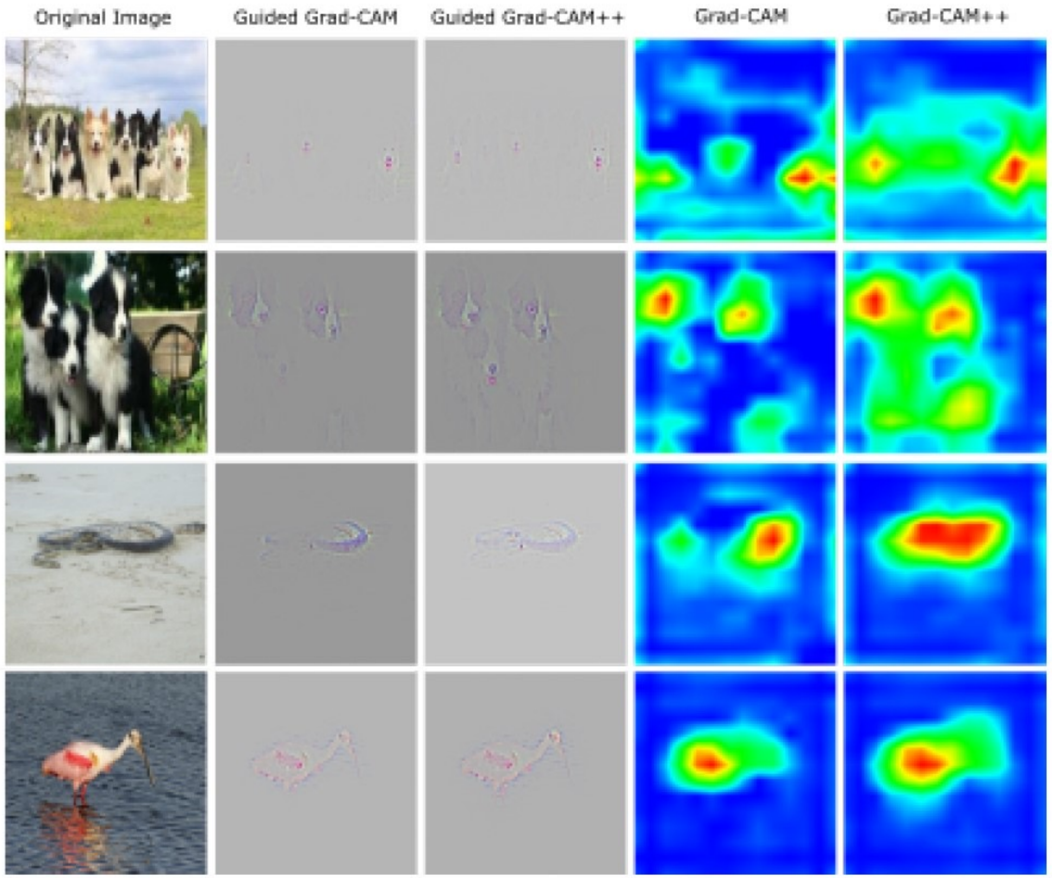
Limitation of Grad-CAM

- Limitation of Grad-CAM
  - Multiple occurrences

    : Grad-CAM fails to properly localize objects in an image if the image contains multiple occurrences of the same class.

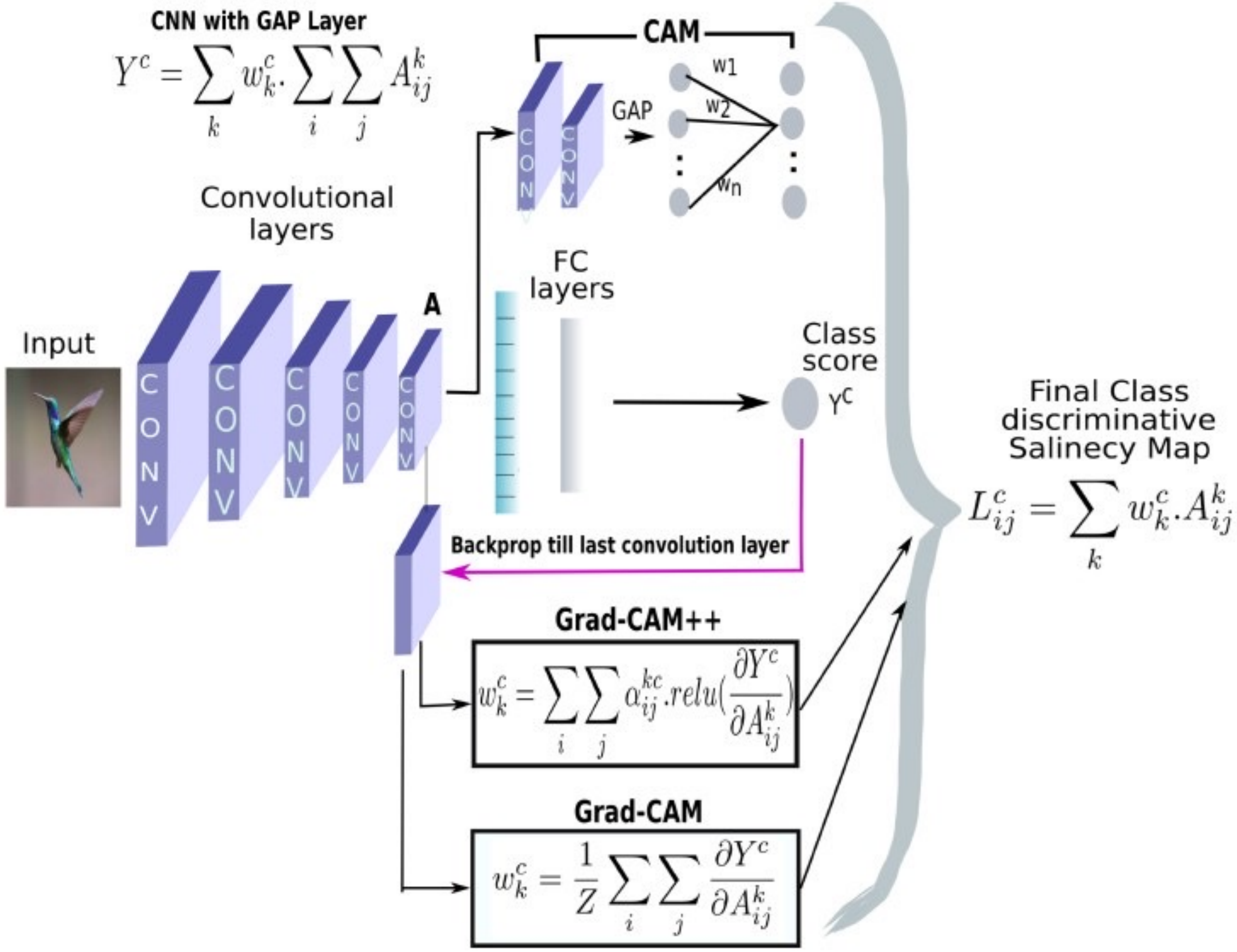
  - Not capturing the entire object

    : An unweighted average of partial derivatives is that often, the localization doesn't correspond to the entire object, but bits and parts of it.



A Chattopadhyay et al. (2017), Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks

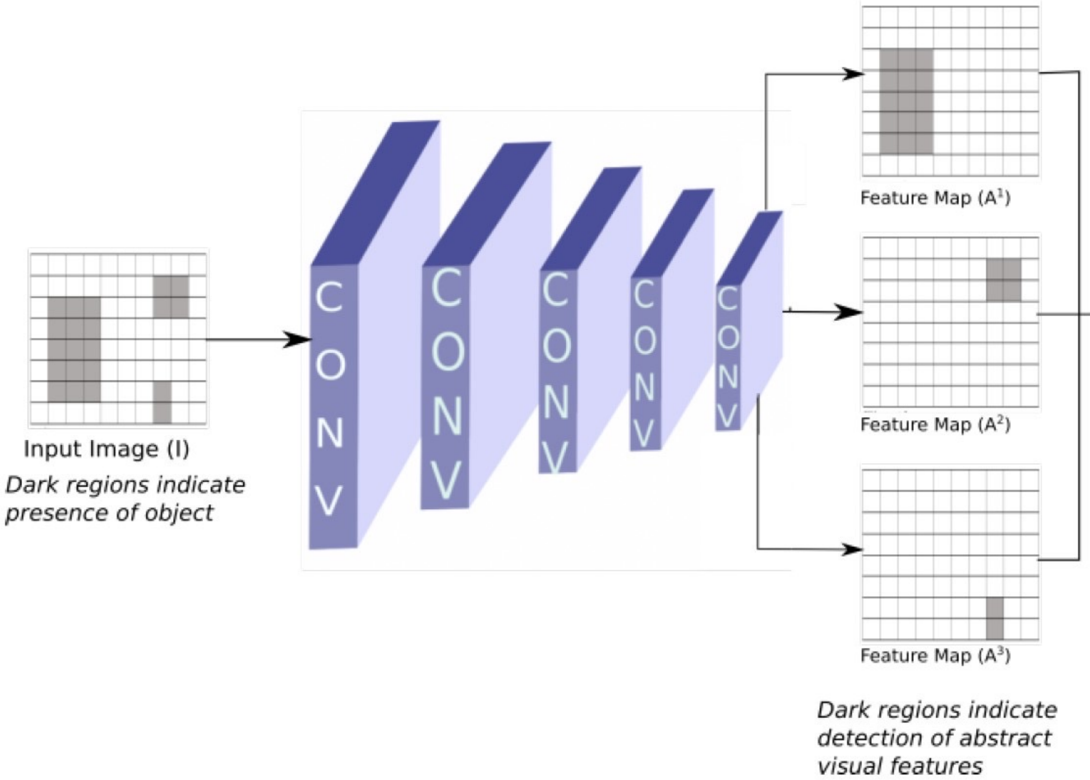
Grad-CAM++



- How to solve the problem?
  - Taking a weighted average of the pixel-wise gradient.
  - If there were multiple occurrences of an object with slightly different orientations or views (or part of an object that excite different feature maps), different feature maps may be activated with differing spatial footprints, and the feature maps with lesser footprints fade away in the final saliency map.

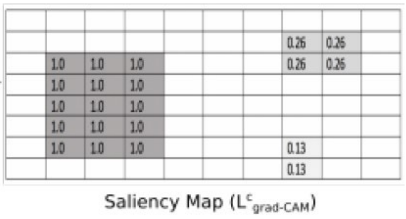
A Chattopadhyay et al. (2017), Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks

Grad-CAM++

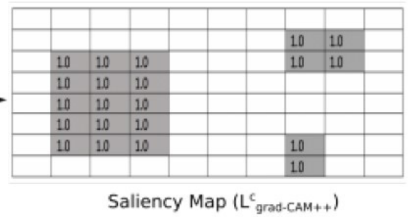


Assumption

$$\frac{\partial y^c}{\partial A^k_{ij}} = 1 \quad if \quad A^k_{ij} = 1$$
$$= 0 \quad if \quad A^k_{ij} = 0$$



$$w^c_1 = \frac{15}{80},\ w^c_2 = \frac{4}{80} \text{ and } w^c_3 = \frac{2}{80}$$



Grad-CAM

$$w^c_k = \frac{1}{Z}\sum_i \sum_j \frac{\partial Y^c}{\partial A^k_{ij}}$$

Grad-CAM++

$$w^c_k = \sum_i \sum_j \alpha^{kc}_{ij}.relu(\frac{\partial Y^c}{\partial A^k_{ij}})$$

$$\alpha^{kc}_{ij} = \frac{\frac{\partial^2 Y^c}{(\partial A^k_{ij})^2}}{2\frac{\partial^2 Y^c}{(\partial A^k_{ij})^2} + \sum_a \sum_b A^k_{ab}\{\frac{\partial^3 Y^c}{(\partial A^k_{ij})^3}\}}$$

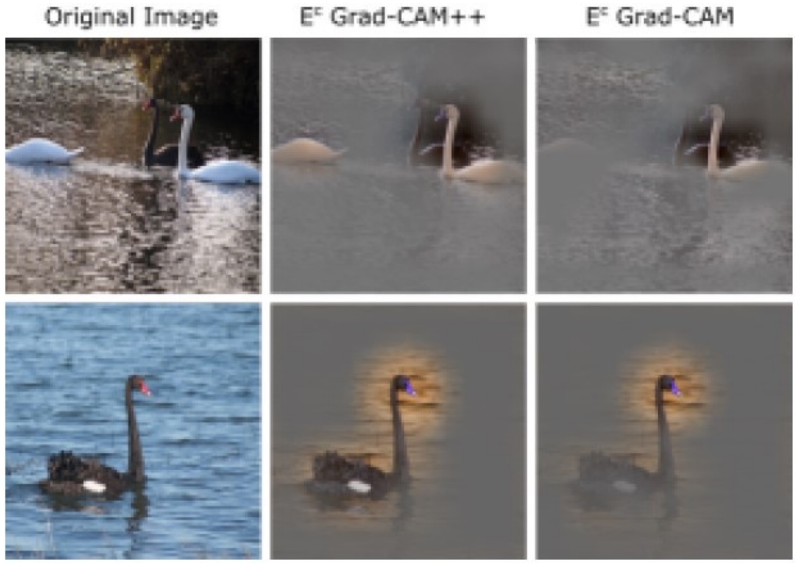
: weighting co-efficient for the pixel-wise gradients for class c and feature map A^k

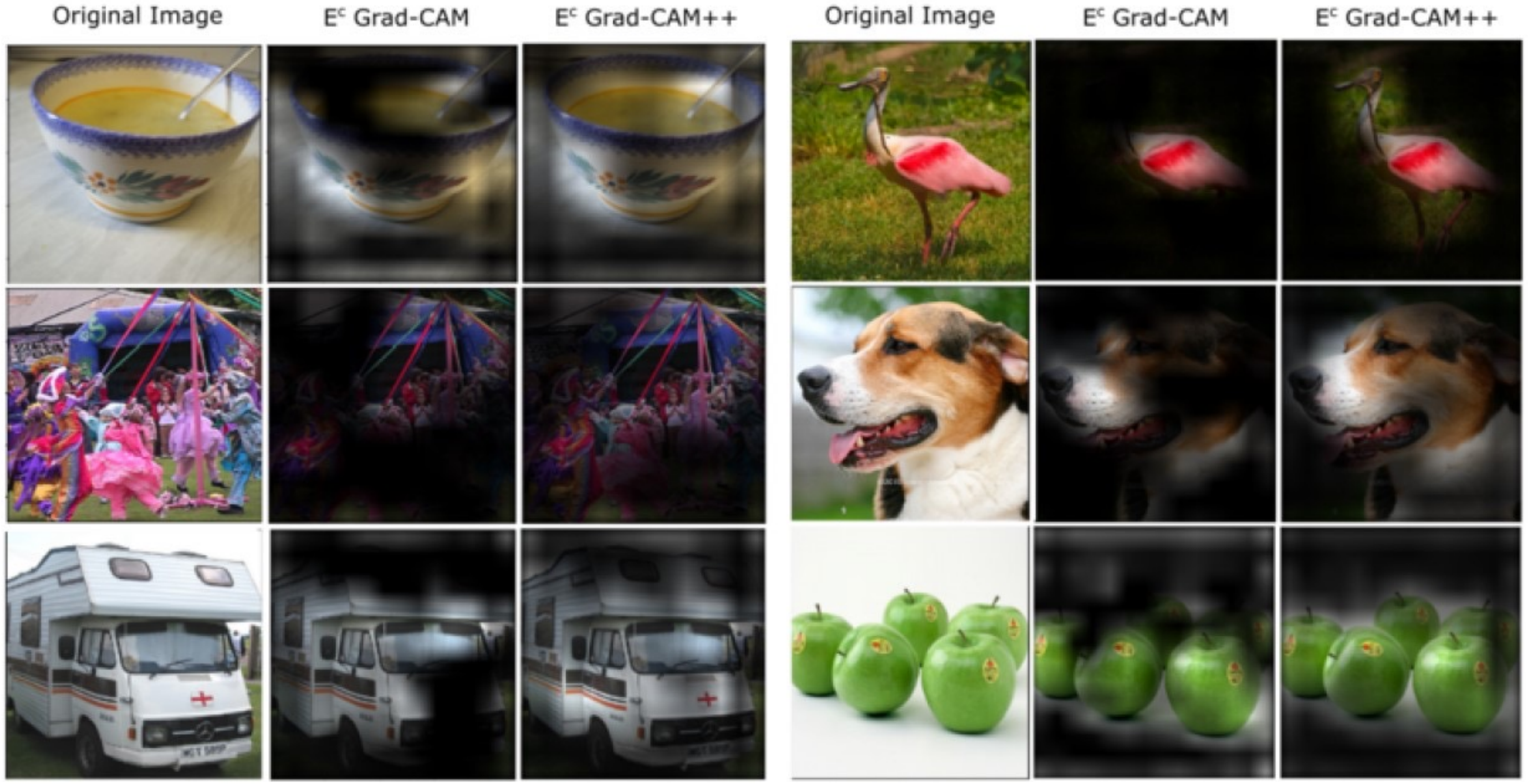
Final Saliency Map

$$L^c_{ij} = \sum_k w^c_k . A^k_{ij}$$

A Chattopadhyay et al. (2017), Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks

Grad-CAM++ Results





A Chattopadhyay et al. (2017), Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks

5. Code Tutorials

Grad-CAM Code

```
# ============================== #
# ==== Grad-CAM main lines ==== #
# ============================== #


# Tabby Cat: 281, pug-dog: 254
score = logit[:, 254].squeeze() # 예측값 y^c
score.backward(retain_graph = True) # 예측값 y^c에 대해서 backprop 진행

activations = feature_blobs[0].to(device) # (1, 512, 7, 7), forward activations
gradients = backward_feature[0] # (1, 512, 7, 7), backward gradients
b, k, u, v = gradients.size()

alpha = gradients.view(b, k, -1).mean(2) # (1, 512, 7*7) => (1, 512), feature map k의 'importance'
weights = alpha.view(b, k, 1, 1) # (1, 512, 1, 1)

grad_cam_map = (weights*activations).sum(1, keepdim = True) # alpha * A^k = (1, 512, 7, 7) => (1, 1, 7, 7)
grad_cam_map = F.relu(grad_cam_map) # Apply R e L U
grad_cam_map = F.interpolate(grad_cam_map, size=(224, 224), mode='bilinear', align_corners=False) # (1, 1, 224, 224)
map_min, map_max = grad_cam_map.min(), grad_cam_map.max()
grad_cam_map = (grad_cam_map - map_min).div(map_max - map_min).data # (1, 1, 224, 224), min-max scaling

# grad_cam_map.squeeze() : (224, 224)
grad_heatmap = cv2.applyColorMap(np.uint8(255 * grad_cam_map.squeeze().cpu()), cv2.COLORMAP_JET) # (224, 224, 3), numpy
```

Code references

1. Github : CAM based methods https://github.com/jacobgil/pytorch-grad-cam

2. Github : Grad-CAM
   https://github.com/PeterKim1/paper_code_review/tree/master/8.%20Learning%20Deep%20Features%20for%20Discriminative%20Localization(CAM)

Reference papers

1. LeCun et al. (1998), GradientBased Learning Applied to Document

2. Springenber et al. (2014), STRIVING FOR SIMPLICITY:THE ALL CONVOLUTIONAL NET

3. B Zhou et al. (2015), Learning Deep Features for Discriminative Localization

4. RR Selvaraju et al. (2016), Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

5. A Chattopadhyay et al. (2017), Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks

Thank you!